DCEL: doubly connected edge list
Contains: vertex list, face list, and edge list

v_s = source
v_t = destination
$f\ell$ = face left
f_r = face right
e_{next} = next e on $f\ell$
e_{prior} = prev e on f_r

stored as ints that = indices in other lists!

$\leq 3n-6$

$\leq 2n-4$
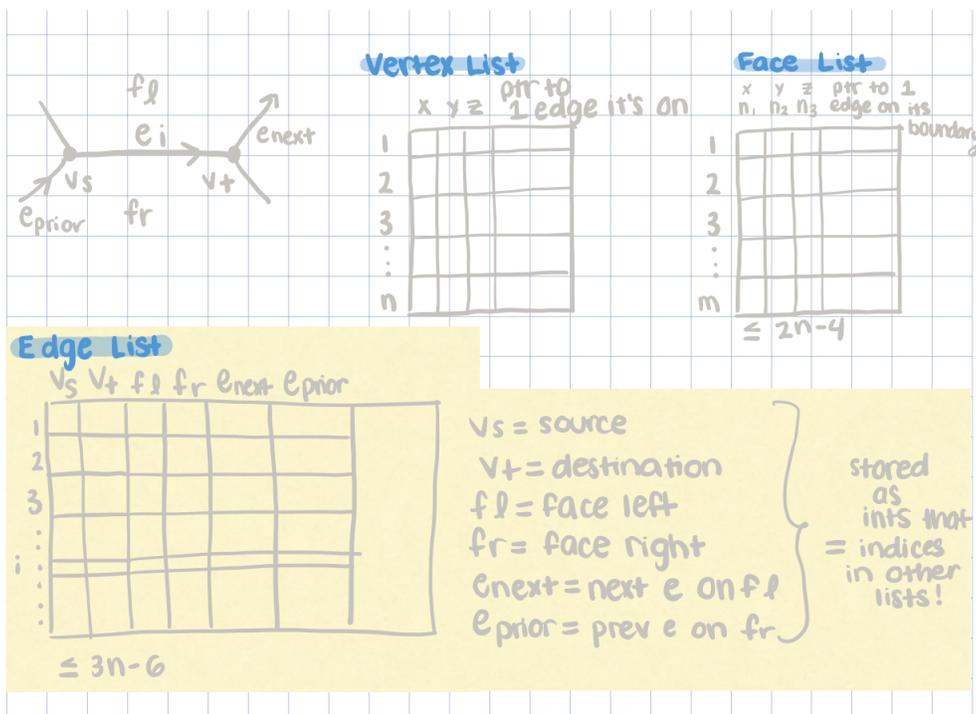
Euler's Theorem:

- Vertices: n

- Faces: $\leq 2n - 4$

- Edge: $\leq 3n - 6$

Ex)
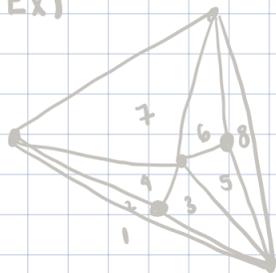
n = 6
e = 12
f = 8

Euler's Theorem
$f + v - e = 2$

n = #v = 6

$f \leq 2n - 4 = 8$ ⎫ equality if ALL
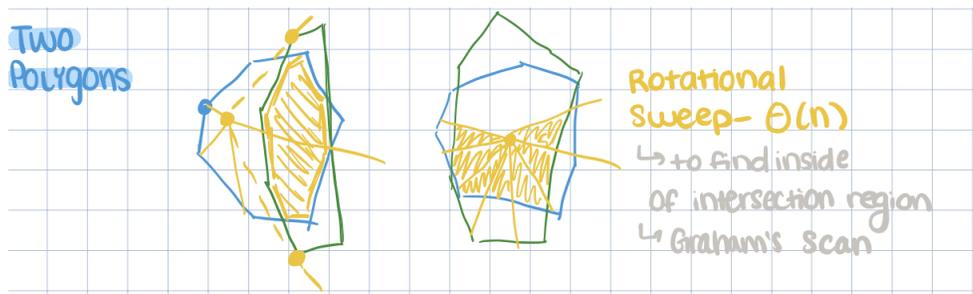$e \leq 3n - 6 = 12$ ⎭ regions (both finite & infinite) are △s
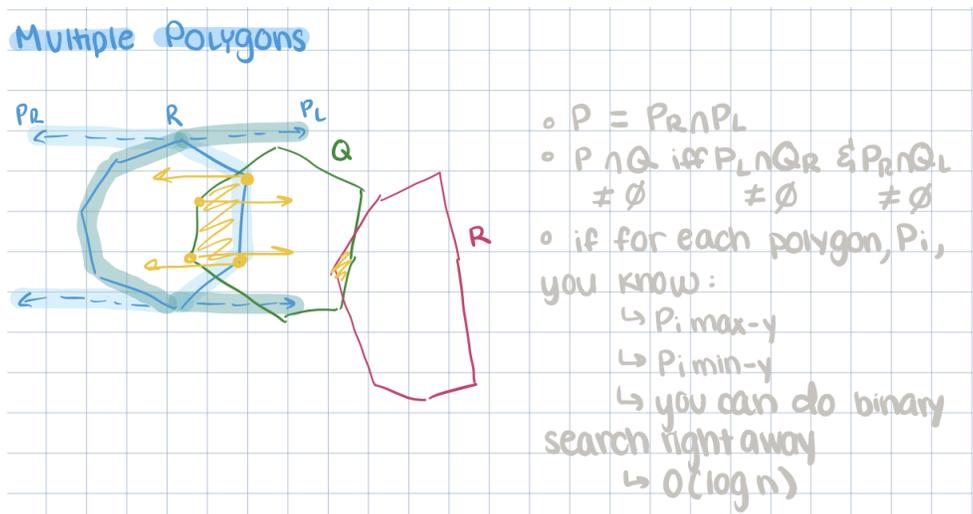
Constant time to walk through!
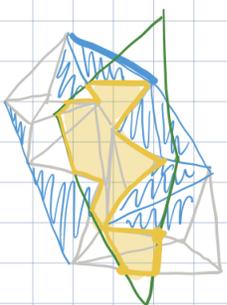
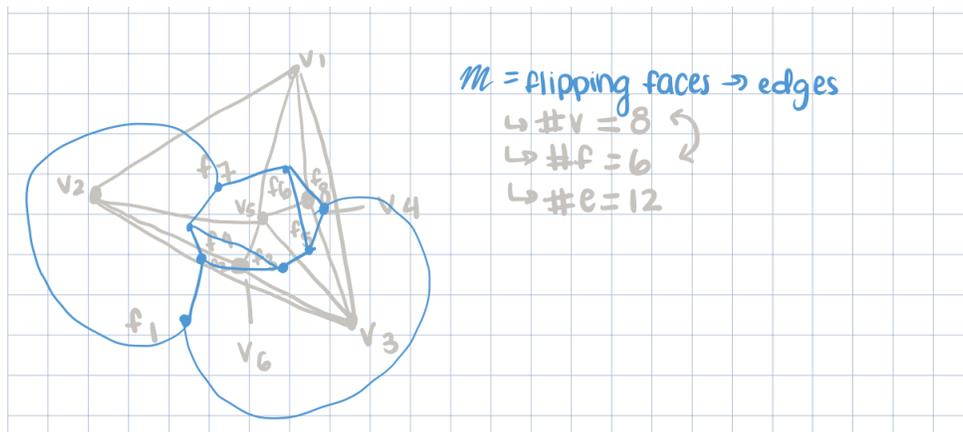○ can walk through in linear time to find intersection region due to DCEL!

Finding Intersection Regions of Polygons:
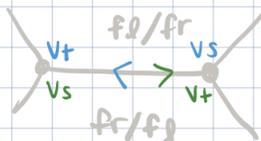## Rotational Sweep (2 Polygons): $\Theta(n)$

Two Polygons

Rotational Sweep- $\Theta(n)$
↳ to find inside of intersection region
↳ Graham's scan

Multiple Polygons:

Multiple Polygons

P_R R P_L

Q

R

○ $P = P_R \cap P_L$
○ $P \cap Q$ iff $P_L \cap Q_R$ & $P_R \cap Q_L$
      $\neq \emptyset$ $\neq \emptyset$ $\neq \emptyset$
○ if for each polygon, P_i, you know:
  ↳ P_i max-y
  ↳ P_i min-y
  ↳ you can do binary search right away
    ↳ $O(\log n)$

3

Graph Dual: Flipping Faces → Edges and Edges → Faces

m = flipping faces → edges
↳ #v = 8
↳ #f = 6
↳ #e = 12

Quad-Edge List
(another way of storing the DCEL)

Quad-Edge List~ each edge in list 4x!

∘ improves coding the DCEL